700 IBM Technical Disclosure Bulletin
30 (1987)Dec., No.7, Armonk,NY,USA

G06F5/06

p357-359

G06F9/46C4

WORKSTATION LOAD LEVELING TECHNIQUE USING BUFFER ALLOCATION

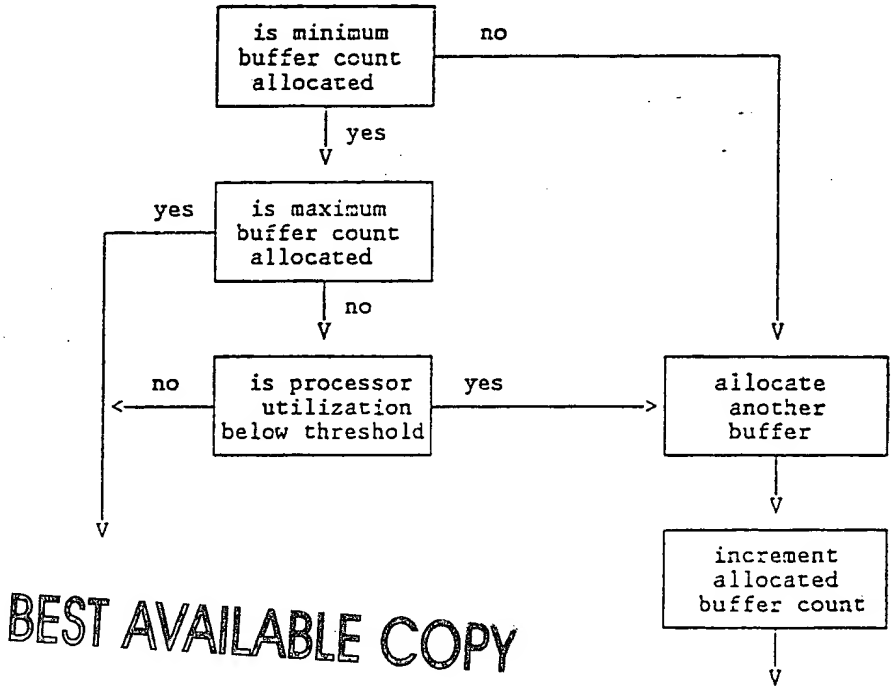

Fig. 1

This technique helps to more effectively utilize a data processor by avoiding unproductive queueing. When work is arriving at a rate which can be handled effectively, work queues remain small. However, when work is arriving at a faster rate, the new work must be added to already long queues. The time spent queueing further reduces the time available for processing. The technique described here helps to eliminate the unproductive queueing time - therefore increasing effective throughput. As a secondary effect, the number of interrupts is also reduced which further increases the throughput.

At a high level, the technique is to artificially limit the number of workstation requests which will be accepted. After a fixed minimum number of buffers are in use, acceptance of additional requests is limited. Specifically, additional requests are accepted only when processor utilization indicates that more input requests can reasonably be handled. This contrasts with a "normal" buffer management scheme which would somehow insure that all required buffers for a "worst-case" situation would be available.

WORKSTATION LOAD LEVELING TECHNIQUE USING BUFFER ALLOCATION - Continued

The technique involves intentionally limiting the number of workstation requests which will be accepted. This limitation is not simply a buffer storage limitation. Instead, this limitation is based upon the ability of the processor to handle additional requests.

More specifically, this technique uses first a small fixed allocation of buffers. Workstation requests will be handled unconditionally as long as they can be accommodated by this small fixed buffer supply. When the fixed supply of buffers is all allocated, additional buffers will be used only when the processor workload is below a specified utilization level. Further, there is also a maximum number of buffers specified. When this maximum number is allocated no more will be used even if the storage and processor time is available (see Fig. 1).

The effect of these limitations is to effectively reduce or eliminate time which would have previously been wasted managing queues. Further, while this technique may reduce the level of service, it is actually greater than it would have been without this technique. Of course, this technique cannot be extended indefinitely. Some number of buffers considerably greater than one is required to effectively handle the parallelism which can be expected. The fixed minimum number of buffers is provided to ensure that the processor is not forced to wait because insufficient work is available.

To implement this technique, a fast, simple method of determining the processor utilization is required. When the processor has nothing else useful to do, it is placed in an idle loop. Within this loop, one instruction increments a counter.

This counter is copied and then set to zero during hardware timer interruptions. A routine which wishes to check the processor utilization needs to only check the copied counter. A large number indicates little processor utilization as the processor spent a lot of time in the idle loop. Conversely, a small number indicates a heavy processor utilization (see Fig. 2).